TITLE: Information Exchange in Support of C2-Interoperability

Information Exchange in support of C2-Interoperability

Freek N. Driesenaar, MSc
Scientist
TNO Phyisics and Electronics Laboratory
P.O. Box 96864
2509 JG The Hague
The Netherlands
Driesenaar@fel.tno.nl

1 Introduction

Large organisations, such as NATO and the armed forces of its member countries, cannot function without the availability of accurate, timely, complete and consistent information. The quality of every decision that is made depends largely on the quality of the information on which the decision is based. This makes information an essential resource for any organisation that must be managed carefully.

Due to the intensified level of co-operation between NATO countries, it has become crucial that information can also be shared *between* armed forces. National forces are deployed ever more often in crisis management situations and (disaster-)relief operations throughout the world, requiring them to work together closely with forces of other countries. Fast and effective collaboration requires a method for information dissemination that is flexible and open.

The need to share information between countries translates directly to the requirement that information can be exchanged between their command & control (C2) systems. For this to be possible, the systems must agree to exchange and interpret information in a standardised (unambiguous) way. In other words: the systems must be interoperable.

This paper focuses on two existing information exchange standards: ADatP-3 (based on formatted messages) and ATCCIS (based on database replication). After describing and analysing both AdatP-3 and ATCCIS separately, the paper compares the two information exchange standards. Ideas are set forward for a unified approach which tries to capture the best of the two worlds and the paper ends with suggestions for future work.

2 Interoperability

Interoperability is defined here as "the ability of two or more systems or components to exchange information and to use the information that has been exchanged" [1]. To explain this concept and to identify which elements are necessary for interoperability, we will examine information exchange as it occurs in different domains (between people and between systems) and we will describe its status quo and how this came to be.

2.1 Interoperability between people

For one person (the provider) to successfully transfer information to another (the receiver), agreements must be made at various levels. First, they must agree upon a medium of communication. If the provider uses writing but the receiver is illiterate, the exchange will fail. If the provider uses speech but the receiver is deaf, again the exchange will fail (although in this case, having the receiver lip-read may solve the problem).

Second, they must agree upon a language. If the chosen medium is speech but the provider speaks in a language unknown to the receiver, there will still be no exchange; that which is spoken may be heard, but it is not understood. The root of the problem lies in the fact that different languages have different vocabularies: they use different words to express the same ideas. This can also occur within a single language, when a speaker uses a jargon that is unknown to the listener. Agreeing upon a language not only entails agreeing upon a vocabulary, but also agreeing upon a common meaning for the words. Even if the provider does speak in a language which is known by both, if both parties attach different ideas to the same words (e.g., what is their definition of "entity"?) they may think they understand one another, while in fact they disagree.

Finally, they must agree upon a common communication procedure. It is no use standardising the format of a request, for example, when in practice the receiver fails to respond to requests because they are not going through the proper channels.

The extent to which these agreements can be made determines the level of understanding that can be achieved between the provider and receiver, and as such, the potential level of interaction between them.

2.2 Interoperability between systems

The agreements that must be made between people are the same agreements that must be made between C2-systems that wish to exchange information. First, they must agree upon a medium, i.e. the type of connection

Paper presented at the RTO IST Symposium on "New Information Processing Techniques for Military Systems", held in Istanbul, Turkey, 9-11 October 2000, and published in RTO MP-049.

will be used to communicate: what type of cable or frequency will physically connect the systems, and what protocol will be used to transport the messages that are sent.

Second, they must agree upon a language that is to be 'spoken' by the systems, i.e. the messages that will be exchanged. Each system has its own native language, which is contained in the structure of the information that is used by that system. For example, the structure may specify that there are clients; that clients have an address and a city of residence; and that clients can place one or more orders. Different systems will generally speak different languages: a 'client' in an order-processing system can be a 'debtor' in a financial administration package and can be a 'lead' in a sales-support system. Therefore, in order to exchange information between systems, it is necessary to create a common frame of reference for the concepts which exist in the individual information structures. In other words, an exchange language must be defined, which describes the messages in terms of syntax (what do they look like) and semantics (what do they mean).

Finally, they must agree upon a set of procedures which regulates the exchange of information: what is the (higher-level) protocol for message exchange between systems, which security considerations must be taken into account, which priorities will be supported, etc.

2.3 Past to present

In the last decade interoperability has become one of the most important issues in system design. This contrasts sharply with the early years, during which there was little need for interoperability. Initially, systems were designed to operate as stand-alone, autonomous units, dedicated towards supporting the work in a particular area or department. Each system had its own form of internal data storage that provided little if any access for external parties. In the few cases that information exchange between systems was required, a dedicated coupling (in the form of a translator) was custom-built.

As technology progressed and the number of systems grew, the need for information exchange increased. It proved infeasible to continue to develop and maintain the increasing number of system-specific couplings. The focus shifted towards finding ways in which couplings could be re-used or could be used to connect multiple systems together. Hardware standards were developed concerning cables and connectors; software standards were developed concerning protocols and services. From the bottom up, the various levels of the OSI-model were filled in.

Now that many technical problems have been solved and boundaries have been pushed back, it is becoming clear that to achieve true interoperability we need some crucial standards. There are different mechanisms available today which allow systems to connect to others, but these do not tell a system how to format and interpret messages that can be sent over the connection. In other words, the medium has been taken care of; the language and the procedures have yet to be worked out.

This setting formed the point of departure for NATO, which was seeing a growing need to interconnect the C2-systems of its member nations. NATO identified the absence of a standardised military language and message exchange protocol that would help its forces to communicate and interact more effectively. To solve this problem, different projects have been initiated over the years to devise a solution. These projects have taken different approaches towards designing an information exchange mechanism, but the two most successful approaches have been the use of formatted messages by ADatP-3 and the database replication approach taken by ATCCIS. Both approaches will be examined in later sections.

3 C2 Information

In order to judge the merit of ADatP-3 and ATCCIS as approaches towards achieving C2 interoperability, we must be clear on what type of information is exchanged between C2 systems. It then becomes possible to indicate to which degree each approach succeeds in supporting specific types of information exchange.

Here we wish to consider two types of C2 information: the actual content and transfer information. Content information is the information that is to be conveyed to a receiver; it is what would normally be written in a letter. Transfer information is the information that determines how the content is to be transferred; it is what would normally be provided on the envelope that contains the letter. Both will be examined in the following subsections.

3.1 Content

As indicated above, content is the information that is being exchanged. As such, this is the information that an exchange language must be able to express. Content comes in three flavours: descriptions, events, and reporting data (for simplicity, we do make a distinction between data and information).

Descriptive data describes the static C2 world; it refers to information that does not change (often) over time. For example, the name and nickname of a unit; the maximum cross-country speed of a Leopard-2 main battle tank; and the location of a town. This type of information can generally be provided ahead of use, in the form of a database or document, but it is sometimes necessary to be able to request it as the need arises.

Event data describes the dynamics of the C2 world; it refers to information that can change (often) over time. For example, the location and status of a unit; the identity of an as yet unidentified person; the sighting of an aircraft; and the available capacity of a field hospital. This type of information can not be provided ahead of

time, but will be reported on a regular basis or as soon as the event occurs.

Finally, *reporting data* is meta-data that provides a context for interpreting description- or event data. For example, the source of the information; the reliability of the source; the credibility of the data; and the time period of validity. This type of information will generally be reported together with the data it refers to.

3.2 *Transfer information*

Transfer information describes how the content is to be exchanged. As such, this is the information that must be used by the exchange medium: it determines how the information is communicated. For example, the identities of the sender and the intended recipient; the priority; the classification; and the type of encryption.

4 Approach 1: ADatP-3 (Formatted messages)

4.1 *Introduction*

ADatP-3 (Allied Data Publication 3) is the name of the publication which documents the NATO Message Text Formatting System (FORMETS); the abbreviation is also widely used to denote that same system. FORMETS specifies the message formats that are to be used in the construction of character-oriented messages that are exchanged between national and NATO authorities and systems. The use of ADatP-3 by all NATO countries has been ratified in STANAG 5500.

The goal of ADatP-3 is to serve as a standard for information exchange in general; not to specifically support exchange between systems. For this reason, ADatP-3 focuses on defining a message standard in which messages are concise, accurate and can be quickly processed by both human operators and automated systems. ADatP-3 specifies only the permitted message formats; it does not make any assumptions concerning the communication medium (although one of the most popular exchange mechanisms for ADatP-3 messages has been ACP127).

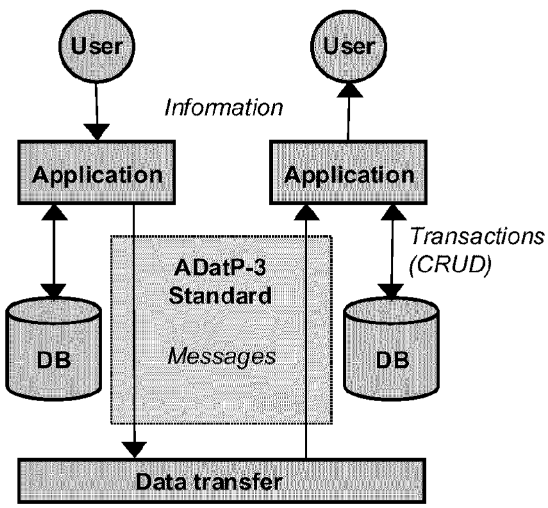

Figure 1 - Information flow between ADatP-3 systems. The shaded area identifies the scope of ADatP-3 work.

The use of ADatP-3 is very straightforward (see Figure 1). A user can transfer information to another user by either writing a message manually, or by generating the message using an automated system. The message can then be sent over any acceptable data transfer mechanism, and after receipt can be processed manually or automatically by the receiver.

4.2 *Exchange language*

ADatP-3 is in fact nothing more than an exchange language. It comprises an artificial, character-based language in which:

- the vocabulary is limited to a collection of codes and words, called fields, which have an unambiguous meaning;
- sentences are limited to certain sequences of fields, which are called sets, in which the position of a field is used to determine its meaning;
- messages are limited to certain sequences of sets, called message text formats (MTFs), in which the position of a set is used to determine its meaning.

The MTF definitions in ADatP-3 are independent of one another; however, MTFs can make use of the same sets, and sets can make use of the same fields. To illustrate the structure of an ADatP-3 message, here is an example:

```
MSGID/ENEMY SITREP/RPVGS/004//
EFDT/040849Z/JUL//
EGROUP/U0004/ORC//
LOCATION/REAL/-/-/-/POINT/32UPC9307//
SOURCE/-/RPV//
TIME/AT/040840ZJUL//
```

In the example, each line is a set, and each set consists of a set identifier (the first word) followed by one or more fields. The first set identifies the MTF that was used; in this case the message is of type ENEMY SITREP.

ADatP-3 messages support primarily the exchange of event data and reporting data. If necessary, description data can be provided in the form of free text, but this has no formal structure and cannot easily be used by automated systems. Transfer data that is supported by ADatP-3 are sender, message type and SIC codes; these can all be contained in the message itself. The format makes no assumptions concerning additional transfer information that may be used by the message transfer mechanism.

4.3 *Advantages*

The ADatP-3 approach has a number of advantages.

First, messages can be *processed independently*. ADatP-3 messages are designed to be self-supporting; they can contain only few references to external sources. As such, an ADatP-3 system does not require messages to arrive in any particular order because it can generally interpret each message in isolation.

Second, ADatP-3 messages are indeed quite *concise*. The formatting allows a lot of information to be provided in a small space.

Third, the message formats are *man-readable*. In part, this is due in part to the choice for an entirely character-oriented format. However, because message- and set headers in the messages provide helpful context information, and because the field-codes adhere to widely used abbreviations, most messages can be read and understood without requiring detailed knowledge of the ADatP-3 format. In fact, even messages that become damaged during transfer may still provide valuable information to a human operator.

Finally, ADatP-3 is a *mature standard* in that a large amount of user-feedback has been obtained with which the format has been improved in iterative steps.

4.4 Disadvantages

The ADatP-3 approach also has a number of disadvantages.

First, ADatP-3 *defines only the syntax* of the exchange language, not the semantics. Field codes are defined in terms of what they abbreviate, but their meaning within a set or the meaning of a set within a MTF are not specified. Although the meaning can often be inferred from the context (see also the first advantage noted above), different interpretations can exist.

Second, ADatP-3 is not always elegantly designed for use in automated systems because of some *minor design flaws*: Some fields permit the use of multiple units of measure; e.g., liquid amounts can be specified in liters or in gallons. Fields are sometimes ambiguous; e.g., a date can be specified either as DDMMYY or YYMMDD. Combinations of fields permit the same information to be specified in different ways; e.g., an armoured infantry unit can be identified by `/ARMD/INF/-/-/` or by `/-/INF/-/ARMD/` or variations thereof. All of these aspects make the development of an ADatP-3 system more complex. Of course, this point relates to the first point.

Finally, ADatP-3 is *not one standard* but a set of standards. The large number of improvements made to the MTFs has resulted in a large number of different versions of ADatP-3, often incompatible with earlier versions. In some cases individual countries have made their own version by adding nation-specific codes and formats, thus adding to the problem.

5 Approach 2: ATCCIS (Database replication)

5.1 Introduction

ATCCIS (Army Tactical Command & Control Information System) is an international study aimed at achieving interoperability between the C2 systems of the participating nations. Thirteen countries are currently active within ATCCIS, and several of these countries are already developing national systems based on the ATCCIS principles.

ATCCIS aims to achieve interoperability by using distributed databases that are synchronised through database replication. The idea is to share information between users by allowing them to write to and read from the same database. However, as a single, centralised database is infeasible in practice, ATCCIS provides multiple nodes in the network with a copy of the shared database, called the replication database, and ensures that changes made to the database at any node are replicated to all other nodes. The ATCCIS solution comprises the following elements:

- an exchange language in the form of a model called the LC2IEDM (Land C2 Information Exchange Data Model), which defines the structure of the shared database;
- an exchange mechanism based on the principles of database replication called the ARM (ATCCIS Replication Mechanism), which allows changes to the shared database to be communicated between nodes; and
- a transfer protocol which is used to transfer the replication messages between the ARMs at the various nodes (this is chosen rather than built; TCP/IP is currently being used).

To illustrate the working of ATCCIS we will examine a simple information flow between two systems. Consider a situation in which two ATCCIS nodes, each comprising of a single application, a geographical information system (GIS), and a copy of the shared database, are connected through a network (see Figure 2). In this example, one user records the movement of a unit using his GIS. This information is translated by the GIS into table updates (creates, updates and deletes) and applied to the replication database (RDB). These database updates are automatically replicated by grouping them in transactions and distributing them using the ARM. On the other end of the line, the transactions are received and applied to the database, and the GIS then translates the updates into information which can be displayed to the second user.

We will now look into the exchange language and the exchange mechanism in more detail.

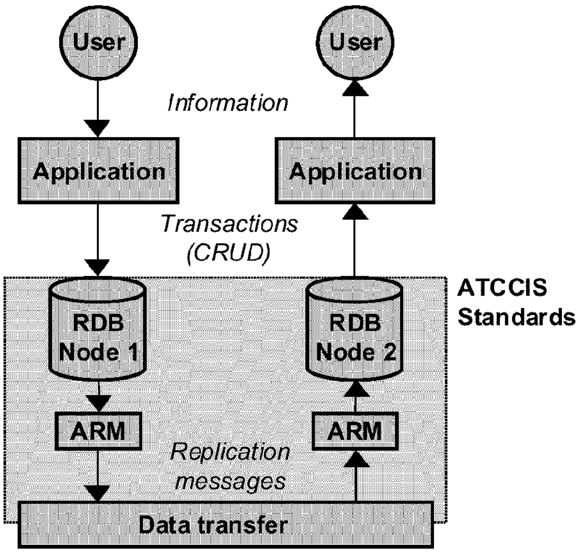

Figure 2 - Information flow between two ATCCIS nodes. The shaded area identifies the scope of ATCCIS work.

5.2 *Exchange language*

The ATCCIS exchange language is defined in a relational datamodel called the LC2IEDM. This model captures the structure of the information that is shared between ATCCIS users. The model is a conceptual model, meaning that it identifies the information concepts that are exchanged without stating how these are to be exchanged.

The scope of the LC2IEDM is the core army C2 information that is exchanged at an international level. Core C2 information refers to the general information concepts that are shared by virtually every unit and cell within the army. For example, the model recognises concepts such as battlefield objects (e.g., units, facilities, terrain features, control features), object characteristics (e.g., location, status, activity), object capabilities, and reports, plans and orders. The model defines these concepts at an international level, meaning that country-specific concepts (such as special naming conventions) are not supported.

However, the LC2IEDM has been developed to be extensible. For example, it is possible to locally add information concepts that are specific to a functional area (e.g., logistics, communications, and engineering) or that are used only by a single nation. In this way, the model acts as the hub of a wheel to which spokes can be added (which is why the model was initially called the ATCCIS Generic Hub Datamodel).

The LC2IEDM provides explicit support for both description data and event data, including the corresponding reporting data. The ARM handles all transfer data.

There are two reasons why the LC2IEDM is indeed the ATCCIS exchange language. First, the replication database contains a direct implementation of this model. This means that the entities and attributes in the relational model have been translated directly to tables and columns in the database. As such, an application that wishes to access or modify the information must do so according to the structures defined in the model, i.e. the access-language is the LC2IEDM. Second, the model directly determines the format of the replication messages that are used by the ARM to exchange information; this is explained in more detail below.

5.3 *Exchange mechanism*

As explained earlier, ATCCIS exchanges information through the use of replication databases. This exchange can be local or remote: local exchange occurs when different users/applications access the same replication database (each replication database can serve multiple clients); remote exchange occurs when information is shared between users on different nodes through replication.

Exchange between replication databases (i.e. remote exchange) is performed by the ARM that must be present on each node. If a modification is made to the local replication database, the associated additions, changes and deletions are sent by the ARM to the ARMs on other nodes in the form of a replication message. If a replication message is received from another ARM, the ARM simply carries out the modifications contained in that message in the local database. In this way, the databases throughout the network are synchronised after each modification. Because modifications are transmitted as changes to tables and records that are identified in the LC2IEDM, the latter model directly describes the structure of replication messages also, again performing in its role as exchange language.

The ARM implements automatic replication based on contracts. A replication contract is an agreement between two users on the information that they will exchange, which is described by four main parts: a data provider, a data receiver, a contract type, and a filter. The use of contracts allows the ARM to work autonomously; modifications are communicated automatically to all nodes that have indicated an interest in the information via a contract (there is no manual trigger required from the user). Furthermore, contracts can be added and removed dynamically as the information requirements change. Note that if there is no contract, then there is no replication.

The ARM permits selective replication through the use of contract types and filters. Given that most tactical networks have bandwidth limitations, it is necessary to reduce the levels of communication as far as possible. Furthermore, due to security considerations it may be desirable to give parties only selected access to information contained in the database. This can be achieved first by using contract types to indicate the type of information that is required: for example, the user may only wish to have information concerning the common

operational picture or information about plans and orders. Next, the information content can be further refined using pre-defined filters that can be parameterised to suit individual preferences: for example, the user may wish to receive only information concerning units in a particular area. As contracts must always be accepted by both provider and receiver, security can be enforced.

All information needed by the ARM to implement automatic, selective replication is stored in the replication database. For this purpose an ARM Management Model (AMM) resides in the database next to the LC2IEDM. The AMM stores information such as the users, the topology of the network (e.g., where are the users located), which pre-defined types of contracts and filters are available, and which contracts and filters which have indeed been defined. The ARM management protocol allows nodes, users, contracts, and flow control to be managed dynamically.

5.4 Advantages

The approach taken by ATCCIS has a number of advantages.

First, the ATCCIS exchange language is *highly consistent*. Because all concepts are contained in a single model that is highly normalised, structures are only defined once. For example, there is only one standard for defining locations or date-time-groups. As another example, the identification of a unit is defined only once in the model and can be re-used wherever necessary.

Second, the ATCCIS exchange language supports *referencing*. So, instead of including for example all information about a unit, one can include a reference to the unit. Of course, this can greatly reduce the size of replication messages.

Third, ATCCIS supports *automatic distribution* of information, as explained in the previous section.

Finally, ATCCIS supports *selective distribution* of information.

5.5 Disadvantages

The approach taken by ATCCIS also has a number of disadvantages.

First, the ATCCIS Exchange Language is *too expressive* to ensure interoperable applications. On the basis of LC2IEDM it is possible to represent, and thus convey, rather complicated information constructs. As a simple example, the LC2IEDM supports report data on event data reported by someone else. The possible constructs are virtually endless and it is certainly possible that applications do not support the same ones.

Second, *event preservation is not explicitly supported*. ATCCIS subdivides events into small segments (e.g., a unit movement is subdivided into a unit segment, a point location segment, a time segment, and the relations between the segments) according to the structure of the LC2IEDM. These segments are then replicated either together or individually, possibly mixed together with segments of other events, and must be regrouped by the application on the receiving end before they can be presented to the user as the initial events. As such, there is no correspondence between events and replication messages; the application must constantly decide whether the latest replicated change will allow it to generate an event or whether it should wait for additional information. This impacts the design of ATCCIS-based applications as well as that of translators that must translate between ATCCIS and other formats (e.g., ADatP-3). It also makes it difficult to implement the filters that can be used in contracts, because a user will generally wish to filter on events rather than on table updates.

Third, *data completeness* is not signalled. It is not always possible to determine whether all database changes relating to a specific event have been received. For example, it is not possible to identify whether all unit locations in a particular plan have been collected. This adds to the problem described above concerning the translation of database updates to user events.

Fourth, ATCCIS replication messages can not be *processed independently*. One reason, of course, is the fact that a replication message can contain data relating to different events. The other reason is that ATCCIS enforces strict referential integrity – meaning that information referenced to should be passed prior to its reference.

Fifth, ATCCIS *replication messages are relatively large*. Replication message syntax does not allow the updating of an individual column in a table record; the entire record must be sent. Next, ATCCIS makes use of technical database keys, which can become very long (e.g., each unit is identified by a unique number of 18 characters). Finally, the structure of the exchange language can cause a small event (e.g., the movement of a unit) to result in many changes to the database, each of which can result in an individual replication message. As such, ATCCIS is not designed to minimise network load, even though it provides support for contracts and filters which reduce the load.

Sixth, there is no support for *varying the quality of service*. All information that is replicated is currently processed with the same level of service: it is sent intact, complete, in order and secure. However, because it is not possible to identify the battlefield event to which a replication message corresponds, it is difficult to assign other service characteristics to messages, such as priority, classification, or time-to-live qualification.

Finally, we observe that ATCCIS is still very much a *standard-to-be*. Little experience has been gained in the practical use of the products, other than what was learned during the few demonstrations held by ATCCIS itself. It

is expected that many lessons learned have yet to be fed back to the standard in order to improve it.

6 ADatP-3 versus ATCCIS

Within the NATO community, ADatP-3 and ATCCIS are viewed as being two completely different approaches towards achieving interoperability between C2 systems. This has resulted in a debate over which of the approaches will best serve for the future. In Table 1 we summarise the results of our analysis of ATCCIS and ADatP-3. Each aspect will be discussed individually below.

Table 1. Comparison between ADatP-3 and ATCCIS. (- : poor, -/+: reasonable, +: good, NS: not supported)

Aspect	ADatP-3	ATCCIS
Consistent	-/+	+
Expressive	-/+	+
Event preservation	+	-
Data completeness	+	-
Independent processing	+	-
Message size	-/+	-/+
Referencing	-	+
Automatic distribution	NS	+
Selective distribution	NS	+
Man-readable	+	-

Consistent: AdatP-3 is not as strict as ATCCIS concerning message syntax and semantics. This is mainly due to the fact that ATCCIS uses a model to derive the syntax and define semantics.

Expressive: ATCCIS is a more expressive approach than ADatP-3, however, ATCCIS is too expressive to enforce interoperability. In ADatP-3, information constructs are constrained to that which can be formulated using the pre-defined MTFs. In ATCCIS, many information constructs are possible and C2-systems are almost bound to differ in the constructs they support, causing (possibly invisible) breaches in or even breaking of interoperability.

Event preservation: ADatP-3 preserves events; ATCCIS does not. ADatP-3 messages contain complete events and can be interpreted in isolation. ATCCIS can replicate events either in a single replication message or using multiple messages, leaving it up to the receiving application to recreate the event for the user.

Data completeness: ADatP-3 signals data completeness, ATCCIS does not. Note that data completeness relates to event preservation: ATCCIS will implicitly signal data completeness, as soon as it preserves events.

Independent processing: In general, AdatP-3 messages can be processed independently, while ATCCIS replication messages can not be processed independently.

Message size and referencing : The amount of data that is physically transferred during information exchange will on average be the same. ADatP-3 messages are concise in comparison with the data that must be replicated when the same information is exchanged within ATCCIS. However, ATCCIS is able to refer to information that has already been sent and only has to send it once, while an ADatP-3 message must always contain all relevant information. In practice, therefore, the amount of information that must be transferred will be comparable (and can be reduced in both cases using compression techniques).

Automatic and selective distribution : ADatP-3 does not support these mechanisms, ATCCIS does. Within ADatP-3, information exchange is initiated by the sender (information-push). ATCCIS, however, allows the receiver to selectively indicate what information he wishes to receive automatically (information-pull).

Man-readable : ADatP-3 messages can be read by human operators; ATCCIS replication messages cannot. ADatP-3 makes use of standard field-codes and uses set identifiers, thus making messages fairly easy to read (although certain message types will require knowledge of the format). ATCCIS replication messages contain table identifiers, numerical database keys (which refer to entities defined in the database) and cryptic mnemonics; their contents cannot be determined without access to the database.

7 Conclusion

We take the view that ADatP-3 and ATCCIS are not completely different approaches, but rather are variations on a common theme. Both can be considered message-oriented solutions: ADatP-3 makes use of ADatP-3 messages, and ATCCIS makes use of replication messages. The main difference between the two is how the messages are generated and how they are processed.

The comparison in the previous section indicates that while neither approach is superior, they complement each other's strengths and weaknesses. This would suggest that a combination might be able to capture the best of both. We therefore come to the following recommendations concerning a unified approach.

7.1 Recommendations for a unified approach

The analysis presented in this paper gives raise to the following recommendations:

- Use a single, unified conceptual model to define the messages of the exchange language (as done in ATCCIS). Allow information structures to be re-

used (e.g., use the same form of unit identification throughout the model). This will result in *consistency* and elegance. Both ADatP-3's MTFs and ATCCIS's LC2IEDM contain many information concepts that can act as starting point for the model.
- Distinguish between description-, event- and reporting data in the model. These can even become separate models. This will keep the model simple and understandable.
- Focus on event data, as this is the most important information that is exchanged between C2 systems. Specify the individual *events* and specify how these are to be mapped to the model and back; leave no room for alternative interpretations. This will limit the *expressiveness* of ATCCIS and ensure and facilitate building interoperable C2-systems.
- Make sure that messages *preserve events* and that messages can be *processed independently*. This will simplify the development of message processing systems.
- Do not require messages to be *man-readable*. Although this was desirable in the past, expect messages to be exchanged between C2 systems only.

From experience obtained in dealing with C2-interoperability matters we would also like to add the following recommendations for the advanced reader:

- Make the conceptual model concrete: do not hide information concepts in abstractions or generic structures. These can be added later when the physical implementation is developed.
- Do not strive to develop a model that can fit on a single page. Allow the model to be multi-dimensional that can be viewed from different angles.
- Consider carefully if the proposed use of the messages (e.g., how will they be filtered and distributed) should affect the structure of the conceptual model. Try to focus only on what will be exchanged at the conceptual level, and include aspects of use at the logical- and implementation levels.

7.2 Future work

In this paper we have looked only briefly at data distribution mechanisms. Although ADatP-3 does not prescribe the use of a particular mechanism, it is primarily suited for point-to-point protocols such as telex and email. ATCCIS bases its own mechanism on automatic and selective replication. When further developing a unified approach it may be worth to consider the following:

- support for point-to-multi-point data distribution – supporting this can result in more efficient data distribution, and may even be essential for use of combat net radios;
- support for different data distribution mechanisms – this may enable a more flexible way to implement automatic information exchange and it may also reduce network load (such as request/reply and publish/subscribe);
- support for various quality of service aspects (such as: priority, assured delivery, confirmed delivery, encryption, compression) – this is especially important in communication critical environments, where the required transmission capacities are close to or even exceed the available transmission capacity, or in cases where the required quality of service exceeds the supported quality of service (for example when an unencrypted classified message is transferred over an insecure data link).
- use of commercially available message oriented middle-ware products, such as: IBM MQSeries, TIBCO TIB/Rendezvous, Talarian MQExpress;
- support other existing interoperability related standards – on the basis of the concrete unified conceptual model and the list of 'events' it may be possible to achieve interoperability using existing standards such as CORBA, COM/DCOM, HLA, and XML. This could enhance the scope of the standard, enable the use of more COTS products, and facilitate the development of applications.

In the end, the unified approach may evolve into an information bus architecture, where C2-systems and/or C2-applications can connect to a C2-network in a 'plug-and-play' fashion.

8 References

[1] IEEE, *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*, New York, 1990

[2] D. Alberts, J. Garstka, F. Stein, *Network centric warfare: developing and leveraging information superiority*, 2nd Edition (Revised), August 1999